

# Planning for IBM DB2 10 for z/OS Upgrade

by John Campbell

## Executive Summary

In the spring of 2010, DB2 10 for z/OS was released to 24 worldwide customers for beta testing. The evaluation focused on regression testing, “out-of-the-box” performance, and additional performance and scalability, as well as other new functions.

Customer experience and feedback about the program have been mainly positive, and most customers who were involved in the program plan to start migration to DB2 10 for z/OS in 2011. An incremental improvement was observed in the effectiveness of the program, in terms of the quality of the issues and problems found, relative to the respective programs for DB2 Version 8 and Version 9. Some customers did very well with regression and new function testing; others provided only limited qualification about what they did and what they achieved.

After the early stages of planning and execution, it is often difficult for customers to sustain the effort required during a six-month period, due to competing business and technical priorities. People, hardware, and time are usually constrained to varying degrees. As of the end of the beta program, no customers were in “true, business production.”

The release of DB2 10 for z/OS provides many opportunities for *price/performance and scalability improvements*. But there is a *tradeoff* in terms of *some increased real storage consumption*. Customers need to carefully *plan, provision, and monitor* their real storage consumption.

The new, 64-bit SQL runtime can provide generous, 31-bit virtual storage constraint relief in the DB2 DBM1 address space. This support provides enhanced vertical performance scalability of an individual DB2 subsystem or DB2 member. It also opens opportunities for further price/performance improvement, through greater use of persistent threads running with the BIND option RELEASE(DEALLOCATE), DB2 member consolidation, and LPAR consolidation.

## Introduction

This paper focuses on the *planning stage* of migrating to IBM DB2 10 for z/OS. The key points of emphasis are:

- ✓ Make sure everyone is educated as to what is needed to ensure project success.
- ✓ Production of a detailed project plan, communicated to all involved, is crucial for success.
- ✓ Some preparation can occur very early, in terms of understanding, obtaining, and installing the prerequisites.

The release of DB2 10 for z/OS was announced on February 9, 2010, and began shipping on March 12, 2010. It was the largest beta test program in the history of DB2 for z/OS.

The information in this paper is drawn from the lessons learned in cooperation with 24 of IBM's largest customers, representing a variety of industries and countries around the world. An extended beta test program started in Q3 2010 and lasted for six months. The program also included 73 parties in vendor programs.

These customers were looking mainly for 31-bit virtual storage constraint relief in the DBM1 address space and all opportunities for price/performance improvement. Other areas of interest included:

- Regression testing (*Be sure to approach regression testing in the order in which you plan to move to production.*)
- “Out-of-the-box” performance
- Additional performance improvements
- Scalability enhancements
- New functions

## Stages of migration

The primary stages of migration to a new version are:

1. Planning
  - » Early stages:
    - Making the decision to migrate
    - Determining what can be gained
    - Planning for prerequisites

- Avoiding incompatibilities
  - Planning performance and storage
  - Assessing available resources
2. Migration
  3. Implementation of the new improvements

Needed application changes can be made over a longer period to make the migration process easier and less costly. Plans for monitoring virtual and real storage resource consumption, as well as performance, are necessary. An early health check, communication of the required changes, and staging of the work will make the project go much more smoothly.

## Highlights of the Beta Test

DB2 10 for z/OS delivers great value by *reducing CPU resource consumption in most customer cases*. IBM internal testing and early beta customer results revealed that, depending on the specific workload, many customers could achieve *“out-of-the-box” DB2 CPU savings of up to 10 percent* for traditional OLTP workloads and up to *20 percent for specific new workloads (e.g., native SQL procedures)*, compared with running the same workloads on DB2 9 for z/OS.

The objective of providing and proving generous, 31-bit virtual storage constraint relief in the DBM1 address space was achieved by the end of the program. This achievement is significant in terms of the enhanced vertical scalability of an individual DB2 subsystem or DB2 member of a data sharing group. We are confident that customers can scale up, in practical terms, the number of active threads by 5 to 10 times to meet their demands.

Further opportunities for price/performance improvement are made possible through the use of persistent threads with the BIND option RELEASE(DEALLOCATE). Examples of using persistent threads include protected ENTRY threads with Customer Information Control System (CICS®), Wait For Input (WFI) regions with Information Management System/Transaction Manager (IMS/TM), and high-performance database access threads (DBATs) for incoming Distributed Data Facility (DDF) workloads.

Another goal was to improve INSERT performance, particularly in the area of universal table spaces (UTS). We wanted to ensure that insert performance for UTS was equal to, or better than, the classic table space types, such as segmented and partitioned. This goal was achieved in most cases.

Hash access was good, provided we hit the smaller-than-expected “sweet spot.” Results for complex queries were also good.

Provided users chose the correct value, the performance of inline large objects (LOBs) was also impressive. Support for inline LOB column values has the potential to save even more on performance by avoiding indexed access to the auxiliary table space. However, it is important to note that the value you choose for the inline LOB value must ensure that most of the LOB column values are 100 percent inline in the base table space.

In the area of latch contention reduction, we focused on the hot latches in DB2 10 for z/OS in such a way that, once we solved the 31-bit virtual storage constraint in the DBM1 address space, enabling you to scale five to ten times, we wanted to be sure there were no secondary issues related to latch contention that would inhibit the vertical scalability of a single DB2 subsystem or DB2 member.

As the beta program progressed, the reliability of, and customer confidence in, DB2 10 for z/OS greatly improved.

Generally speaking, online transaction processing (OLTP) performance improvements were as predicted. We were aiming for a target of 5 percent to 10 percent reduction in CPU resource consumption for most traditional OLTP workloads. During testing, several customers ran benchmarks showing that such reductions could be achieved. However, in cases where the transactions consisted of a few very simple SQL statements, the 5 percent to 10 percent target was not achieved.

This is where the increase in package allocation cost outweighed the improvement in SQL runtime optimization. However, we did identify some steps that can be taken to improve this. We have delivered an Authorized Program Analysis Report (APAR) to reduce package allocation cost. It is also possible to mitigate this situation by making more use of persistent threads with the BIND option RELEASE(DEALLOCATE).

Another issue was single-thread BIND/REBIND performance. Even in Conversion Mode (CM), the performance, in terms of CPU resource consumption and elapsed time, was degraded. One reason for this result was that in DB2 10 for z/OS the default for access plan stability is EXTENDED. Also, DB2 10 for z/OS uses indexed access, even in CM, to access the respective DB2 Catalog and Directory tables.

Another area where we had mixed results was SQL Data Definition Language (DDL) concurrency. We had hoped that by restructuring the DB2 Catalog and

Directory to introduce row-level locking, remove hash link access, and more, we could improve concurrency when running parallel SQL DDL and parallel BIND/REBIND operations. The concurrency improvement was eventually achieved for parallel BIND/REBIND activity. Although it also helped in some cases with SQL DDL, most customers will still have to run SQL DDL activity single-threaded.

The final issue was access path lockdown. Two new options in DB2 10 for z/OS, APREUSE and APCOMPARE, enable you to generate a new SQL runtime while in most cases keeping the old access paths. Unfortunately, there were some issues with the underlying OPTHINTS infrastructure inherited by DB2 10 for z/OS, which is used by APREUSE and APCOMPARE. The introduction of APREUSE and APCOMPARE was delayed until these issues were addressed. These features are now available in the service stream via APARs, and their use is strongly recommended.

In general terms, the results of the beta program were mainly positive customer experiences, and we received good feedback about the program. A majority of customers in the beta program plan to start migrating to DB2 10 for z/OS in 2011. We observed incremental improvement in the program over what we experienced with the DB2 8 and DB2 9 for z/OS programs.

There was really no “single voice” or message across the customer set. We saw significant variation in terms of customer commitment and achievement. A small subset of customers did a very good job on regression and new function testing and provided good feedback. Others, due to limited resources, provided only limited qualification about what they were going to do and what they were able to achieve.

It is worth keeping in mind, for those who have never been involved in a Quality Partnership Program (QPP)/beta program, that it can be a challenge for customers to sustain the effort over a six-month period, due to competing business and technical priorities as well as constraints on people, hardware resources, and time.

By the end of the program, no customers were in true, business production. But we also need to appreciate that a QPP/beta program is *not* the same as an Early Support Program. We continue to develop and test the DB2 for z/OS product as the program progresses.

One of the benefits of DB2 10 for z/OS is that it provides many opportunities for price/performance (cost reduction) improvements. It is a major theme of this release. In discussions with customers, these opportunities for price/performance improvement are most welcome.

Also keep in mind that customers can be intimidated by some of the marketing “noise” about improved price/performance, often because of the raised expectation level of their respective CIOs. But in some cases, it is because when they run their own workloads, they do not see the anticipated improvements in CPU resource consumption and elapsed time performance. Many customers saw big improvements for certain workloads, while for other workloads, they saw little, if any, improvement.

Also note that if you have small test workloads that are untypical of the total mixed workload running in production, this can skew expectations on savings—either positively or negatively. Once DB2 10 for z/OS is in production, the results with the full, mixed workload may differ. We found that some measurements and quotes were overly positive and should be ignored.

A remaining question is: “How do you extrapolate from a small workload and project what the savings would be for the total, mixed workload in production?” Estimating with accuracy and high confidence is not practical, or possible, without proper benchmarking using a workload that truly represents production. Most customers reported incremental improvement over the DB2 8 and DB2 9 for z/OS programs.

Overall, most tests identified opportunities for price/performance (cost savings) improvements, which is the major theme of this release. Some customers reported big improvements in CPU and elapsed time reduction for certain workloads, while others did not. Keep in mind that smaller workloads may skew expectations on savings.

### ***Summary of results***

The DB2 10 for z/OS beta program confirmed improvements in the following areas:

- ✓ 31-bit virtual storage constraint relief in the DBM1 address space
- ✓ Insert performance
- ✓ Hash access good when hitting the smaller-than-expected sweet spot
- ✓ Complex queries
- ✓ Inline large objects (LOBs) and structured large objects (SLOBs)
- ✓ Latch contention reduction
- ✓ Quality of problems and issues found
- ✓ Reliability and confidence as program progressed